Closest Pair Experiment Analysis
Curso : Algoritmos y Complejidad
Código estudiantil: 200156522
Fecha : 21 de noviembre del 2022
Alumna : Natalia Martínez Díaz
Profesor: Misael Díaz Maldonado

Abstract

This document analyses through a java implementation the behavior of two different algorithms that generate n coordinates and determines among all the coordinates generated the pair of coordinates that have the smallest distance between them. The algorithms are known as ”Brute Force” and ”Divide and Conquer”; the main difference between them is that in Divide and Conquer a recursive function is implemented to make it more efficient. The objective is to prove that the algorithm Divide and Conquer is in fact more effective than by Brute Force. For the analysis 16 tests are performed for each algorithm to collect relevant data like the n input, the amount of iterations and the time execution.

1 Introduction

When there are many positions displayed, it is difficult to directly identify from what point to another is the minimum distance. The purpose of this experiment is to analyse two different ways to find the closest pair of coordinates between n amount of coordinates. This algorithms are known as By Brute Force and By Divide and Conquer. By analysing the complexity of each algorithm we will be able to identify which one is more effective and why.

2 Problem Definition

Finding the closest pair of coordinates becomes complex when the amount of coordinates is a really big number. It would take a lot of time to compare every single coordinate to the other coordinates. By proving that the Divide and Conquer method is in fact more effective than by Brute Force we have found a way of finding the closest pair without comparing every single coordinate.

3 Methodology

Java Netbeans was the software used to the program in Java. The program is able to generate n amount of coordinates, sort the coordinates in ascending order depending on the x value. To find the closest pair there are two methods that we are testing:

1. By Brute Force:

    - By brute force two for loops are used to compare every single coordinate with the other coordinates. The distance of the coordinates compared is calculated and to determine which is the closest pair there is a conditional inside the for loops.

2. By Divide and Conquer:

    - The function brute force is also part of the solution of the Divide and Conquer method. The difference is that by Divide and Conquer there is a an additional recursive function that separates the ArrayList of coordinates into many sub Arraylist. When the size of the arraylist is less or equal to 3, the brute force function is executed. Also, the purpose of this separation is to find the group that contains the smallest distance and with that distance create another sub array that only contains coordinates that are that distance away or smaller from the mid coordinate.

To save the amount of coordinates (n), amount of iterations, and time execution we used an Array for each mentioned. Also, the program also generates a file with the information collected after execution (Amount of numbers, amount of iterations, and time execution) to be able to create the graphs and tables shown on the results section. The graphs on the result section were generated by a Python Program developed by Misael Díaz [1].

4 Results

Brute Force:

N	Comparisons	Time execution (NanoSeconds)
8	79	1123909
16	283	86490
32	1113	220025
64	4365	642455
128	17086	1506843
256	67217	1301010
512	266071	5711350
1024	1057478	11931927
2048	4214252	51688149
4096	16821190	199356279
8192	67205113	678447227
16384	268644271	2686747416
32768	1074192336	10690572578
65536	4295934201	51441695018
131072	17181934043	305260394975
262144	68723869004	1660029726918

Table 1: Brute Force Results

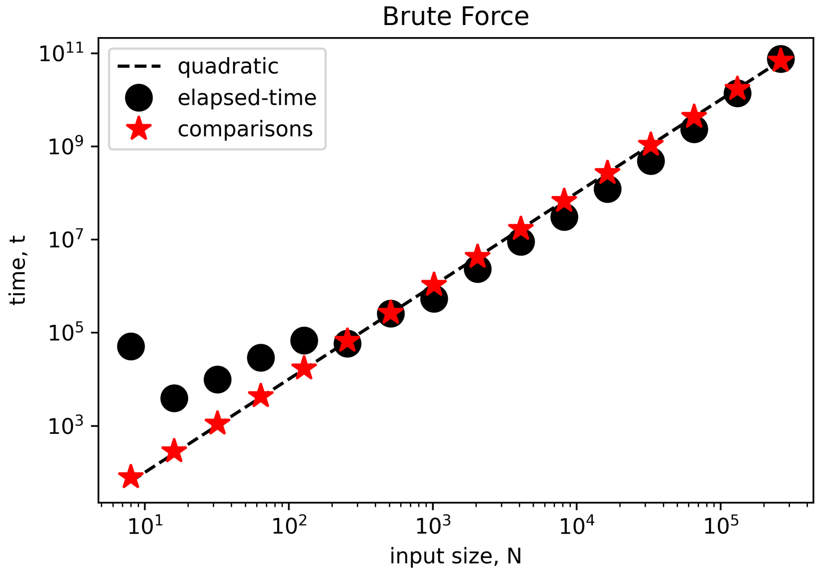

Figure 1: Brute Force Graph

Divide and Conquer:

N	Comparisons	Time execution (NanoSeconds)
8	44	1046945
16	72	94924
32	127	270067
64	258	546082
128	539	616102
256	1088	793045
512	2204	1168157
1024	4482	2280777
2048	8944	5477303
4096	17890	9259060
8192	35738	10542270
16384	71479	16714209
32768	142807	24026676
65536	286380	47108325
131072	573244	120369049
262144	1147391	347920057

Table 2: Divide and Conquer Results

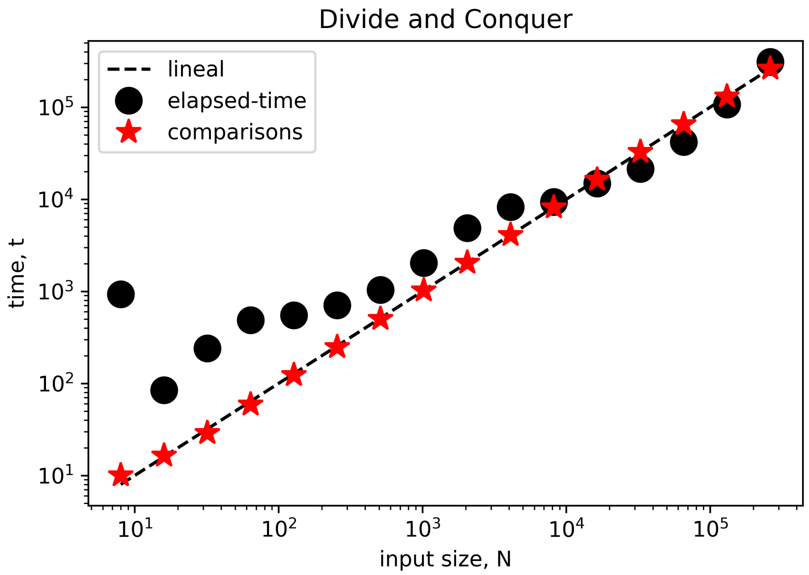

Figure 2: Divide and Conquer Graph

5 Discussion

The results of these tests demonstrate that the Divide and Conquer algorithm is more efficient than the Brute Force. Firstly, looking at Figure 1 it describes in a graph the behavior of the Brute Force. It shows that it has a quadratic complexity, looking at the y axis we can see how the time of execution and the amount of iterations increase rapidly. On the other hand, looking at Figure 2 it has a linear complexity, unlike the Brute Force graph, the y axis maximum value is much smaller meaning the time of execution and the amount of iterations increments are smaller. Also, comparing both tables, it shows that the number of iterations for the Brute Force are greater than the number of iterations for the Divide and Conquer.

6 Conclusions

To conclude, the purpose of this experiment was to demonstrate that the Divide and Conquer method is much more effective than the Brute Force method. The Brute Force method works just fine when there is not many coordinates but when the amount of coordinates increases it becomes way more complex. The Divide and Conquer algorithm simplifies this problem by dividing the amount of coordinates into different groups. To improve this experiment next time we can try this experiment with graphs, applying it more to reality.

References

[1] Misael Diaz Maldonado. *loglogPlot.py*. 2022. URL: //github.com/misael-diaz/computer-programming/%20blob/main/src/io/java/loglogPlot.py%22.